



(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
 17.09.2003 Bulletin 2003/38

(51) Int Cl.7: **G11C 16/10**

(21) Application number: **02425152.2**

(22) Date of filing: **14.03.2002**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE TR
 Designated Extension States:
AL LT LV MK RO SI

• **Redaelli, Marco**
 20048 Carate Brianza (MI) (IT)
 • **Villa, Corrado**
 20050 Sovico (MI) (IT)

(71) Applicant: **STMicroelectronics S.r.l.**
 20041 Agrate Brianza (Milano) (IT)

(74) Representative: **Pezzoli, Ennio et al**
c/o Maccalli & Pezzoli S.r.l.,
Via Settembrini, 40
20124 Milano (IT)

(72) Inventors:
 • **Bernardi, Oreste**
 20044 Bernareggio (MI) (IT)

(54) **A non-volatile memory device**

(57) A non-volatile memory device (120) is proposed. The non-volatile memory device includes a flash memory (205) and means (225) for executing external commands, the external commands including a first subset of commands for accessing the flash memory directly; the memory device further includes a program-

mable logic unit (245) and means (250) for storing program code for the logic unit, the external commands including a second subset of at least one command for causing the logic unit to process information stored in at least one portion of the flash memory under the control of the program code.

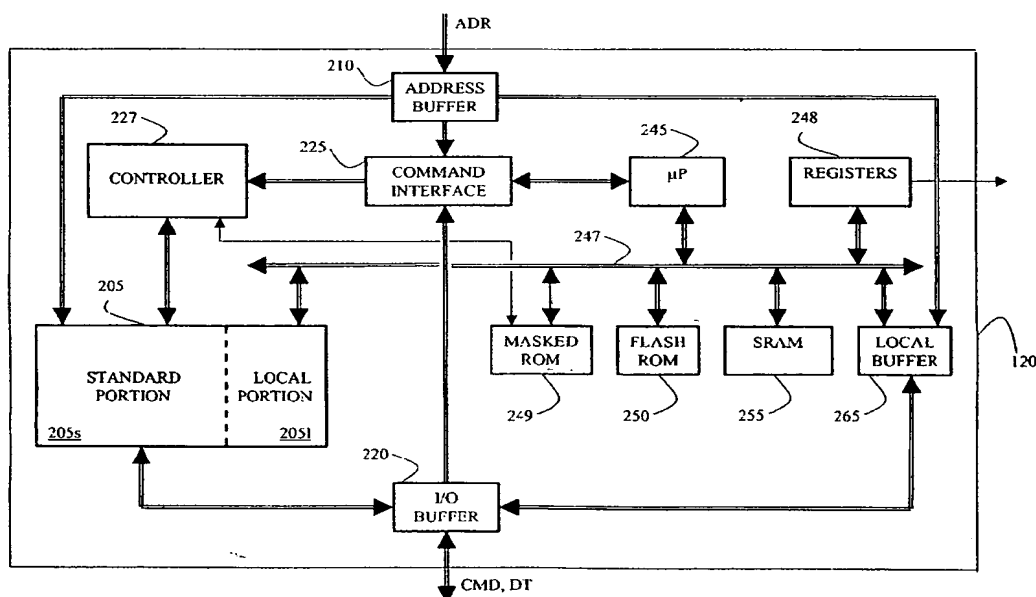


Fig.2

Description

[0001] The present invention relates to a non-volatile memory device.

[0002] Non-volatile memory devices store information, which must be preserved even when a power supply feeding the memory device is off. These memory devices are commonly used as external peripherals (by a processing unit) in several applications.

[0003] A particular type of non-volatile memory device is a flash E²PROM (or simply flash memory). A flash memory consists of a matrix of memory cells, each one formed by a single transistor. This results in a very simple structure of the flash memory, which can be manufactured at low cost and with high density. As a consequence, the flash memory is well suited to a number of end-product applications, such as mobile telephones, automotive devices, digital still cameras, hand-held computers, and the like.

[0004] A drawback of the flash memories is that they must be erased in blocks. The smallest dimension of the block that can be used in practice at a reasonable cost amounts to a few Kbytes. However, the resolution so achieved is insufficient for several applications requiring the ability to modify information at word, byte or even bit level. As a consequence, complex Systems on Chips (SOCs) embedding both a flash memory and an E²PROM (which can be erased a bit at a time) are often required.

[0005] A different solution known in the art is that of using only the flash memory for storing both persistent information (such as program code) and information that needs to be updated (such as data). The data stored in the flash memory is managed by a software program, which controls and organizes the flash memory so as to emulate the behavior of an E²PROM. The data management software is stored in the flash memory, and runs on the processing unit that employs the flash memory as an external peripheral; the flash memory has a multiple banks architecture, so that instructions of the data management software may be fetched and executed from one bank while another bank is being programmed or erased. In this way, the capacity of the non-volatile memory device is increased (for the same cost) or its cost is reduced (for the same capacity).

[0006] A drawback of the solution described above is that the running of the data management software involves an overload of the processing unit; as a consequence, the performance of a whole system (including the flash memory and the processing unit) is reduced.

[0007] Moreover, a large amount of data must be transmitted (typically on a communication bus) between the flash memory and the processing unit. The high traffic on the communication bus increases the power consumption of the system. This drawback is particular acute in portable systems (such as mobile telephones), which are supplied by a battery.

[0008] It is an object of the present invention to over-

come the above-mentioned drawbacks. In order to achieve this object, a device as set out in the first claim is proposed.

[0009] Briefly, the present invention provides a non-volatile memory device including a flash memory and means for executing external commands, the external commands including a first subset of commands for accessing the flash memory directly; the memory device further includes a programmable logic unit and means for storing program code for the logic unit, the external commands including a second subset of at least one command for causing the logic unit to process information stored in at least one portion of the flash memory under the control of the program code.

[0010] Moreover, the present invention also provides a data processing system including the non-volatile memory device.

[0011] Further features and the advantages of the solution according to the present invention will be made clear by the following description of a preferred embodiment thereof, given purely by way of a non-restrictive indication, with reference to the attached figures, in which:

Figure 1 is a diagrammatic representation of a processing section of a mobile telephone in which the non-volatile memory device of the invention can be used; and

Figure 2 shows a schematic block-diagram of the non-volatile memory device.

[0012] With reference in particular to Figure 1, a processing section 100 of a mobile telephone is depicted. The processing section 100 is formed by several units, which are connected in parallel to a communication bus 105. In detail, a microprocessor (μ P) 110 controls operation of the processing section 100, and a SRAM 115 is used directly by the microprocessor 110 as a working memory.

[0013] A non-volatile memory device 120 (described in the following) stores the program code executed by the microprocessor 110. Moreover, the non-volatile memory device 120 is used for storing data that needs to be preserved when a power supply feeding the processing section 100 is off. For example, the non-volatile memory device 120 stores a telephone book and the text of SMSs that have been sent and received by the mobile telephone.

[0014] Further peripheral units are connected to the bus 105 (by means of respective interfaces). Particularly, an input unit consists of a keypad 125 and an output unit consists of a display 130. Moreover, the processing section 100 includes an analog-to-digital (A/D) converter 135; the A/D converter 135 is used to interface the processing section 100 with external units, such as a Radio Frequency (or RF) section, a microphone, and a loudspeaker.

[0015] Similar considerations apply if the processing

section is used in different applications (for example, in a digital still camera, in a music player, in a hand-held computer or in a voice recorder), if the processing section has a different architecture or includes other units, if it is replaced with an equivalent data processing system, and the like.

[0016] Considering now Figure 2, the non-volatile memory device 120 is integrated in a single chip of semiconductor material. The non-volatile memory device 120 includes a flash memory 205, which is formed by a matrix of memory cells (each one consisting of a floating gate MOS transistor). The flash memory 205 is partitioned into a plurality of banks made in respective insulated wells of the chip; for example, the flash memory 205 consists of 28 banks each one with 512k memory cells. Each bank is formed by 8 blocks (or sectors) of 32k memory cells; all the memory cells of a block must be erased at the same time (with each block that may be erased individually).

[0017] A word (typically of 16 bits) is simultaneously processed by the flash memory 205. The flash memory 205 has a multiple banks architecture, which provides a dual operations feature. In this way, while programming or erasing in one bank, read operations are possible in another bank with zero latency (only one bank at a time is allowed to be in program or erase mode); if a read operation is required in a bank that is programming or erasing, the program or erase operation is suspended. The combination of these features means that read operations on the flash memory are possible at any time.

[0018] The flash memory 205 is logically partitioned into a standard portion 205s and a local portion 205l (each one consisting of one or more different banks). The standard portion 205s stores persistent information (such as program code), while the local portion 205l stores information that needs to be updated (such as data). The dual operations feature of the flash memory 205 allows the program code to be executed from the standard portion 205s while the local portion 205l is programmed or erased.

[0019] An address buffer 210 is used as an input structure for receiving an external address ADR, which identifies a location of the flash memory 205 storing a word. In the example at issue, the flash memory 205 is formed by 128 banks each one with 512k memory cells, that is $512/16=32k$ words of 16 bits; the flash memory 205 then stores $128 \cdot 32 \cdot 1024 = 2^7 \cdot 2^5 \cdot 2^{10} = 2^{22}$ words; as a consequence, each location of the flash memory 205 is identified by an address ADR of 22 bits. Moreover, an I/O buffer 220 receives information from the microprocessor external to the non-volatile memory device 120; the information consists either of external commands CMD for the non-volatile memory device 120 or data DT to be written into the flash memory 205; the I/O buffer 220 is further used to provide data DT read from the flash memory 205 to the external microprocessor.

[0020] The address ADR from the buffer 210 and the

data DT from the buffer 220 are supplied to the flash memory 205 directly. The address buffer 210 and the I/O buffer 220 are further coupled with a command interface 225. The command interface 225 interprets the commands CMD received from the I/O buffer 220. The command interface 225 accordingly actuates a controller 227, which is typically implemented with a state machine; the state machine 227 provides a sequence of control signals causing the flash memory 205 to execute the operations required by each command CMD.

[0021] The command interface 225 also communicates with a microprocessor (μP) 245 embedded in the non-volatile memory device 120. Several units are connected in parallel to the embedded microprocessor 245 through a communication bus 247. In detail, a series of high-speed internal registers 248 stores information that is used by the embedded microprocessor 245 directly. Particularly, one register 248 is used to store status information, indicative of an operative condition of the non-volatile memory device 120, and another register 248 is used to store configuration information, indicative of the dimensions of the standard portion 205s and of the local portion 205l of the flash memory; the status register and the configuration register are accessible from the outside of the non-volatile memory device 120 directly.

[0022] A masked ROM 249 is hardwired with preprogrammed instructions for the embedded microprocessor 245. A further ROM 250 (typically implemented with a small flash memory) stores program code controlling operation of the embedded microprocessor 245, which program code is loaded from the outside under the control of the state machine 227. A SRAM 255 (for example, storing 8k words) is used directly by the embedded microprocessor 245 as a working memory.

[0023] The embedded microprocessor 245 is further coupled with the local portion 205l of the flash memory and with a local buffer 265 (implemented with a volatile memory, such as a SRAM storing 1k words); the local buffer 265 is directly mapped on the same address space of the lowest area of the local portion 205l. The local buffer 265 is input a portion of the address ADR from the buffer 210; particularly, the local buffer 265 receives the 10 less significant bits of the address ADR, which are required to identify one of its $1k=2^{10}$ locations. The local buffer 265 is coupled with the I/O buffer 220, in order to receive words to be written into the local buffer 265 and control signals for managing its operation; the I/O buffer 220 is further used to provide a word read from the local buffer 265 to the external microprocessor.

[0024] The command interface 225 supports the most common commands of a standard flash memory. Particularly, a Word Program command and a Word Read command are used to program and to read, respectively, a selected location of the flash memory 205, a Block Erase command is used to erase a specific block of the flash memory 205, an Autoselect command is used to read a manufacturer code and a device code identifying

the flash memory 205, a Read CFI command is used to read a Command Flash Interface defining the specifications of the flash memory 205, and the like.

[0025] Moreover, the command interface 225 supports additional commands. Particularly, a Start command and a Suspend command are used to start and to suspend, respectively, operation of the embedded microprocessor 245. A Reset command forces a hardware reset of the embedded microprocessor 245. A Read/Write Buffer command is used to switch the non-volatile memory device 120 to a local mode of operation, wherein the external microprocessor can access the local buffer 265 (temporarily mapped on the address space of the local portion 205I of the flash memory); a Read Flash command is used to switch the non-volatile memory device 120 back to a standard mode of operation, wherein the external microprocessor accesses the whole flash memory 205 (assuming that the embedded microprocessor 245 is idle). A Code Erase command and a Code Program command are used to erase and to program, respectively, the flash ROM 220 (in order to update the program code stored therein). Finally, a Configuration command updates the content of the configuration register.

[0026] The non-volatile memory device 120 described above is used as a standard flash memory by the external microprocessor. Particularly, assuming that the non-volatile memory device 120 is in the standard mode of operation (local buffer 265 non-accessible from the outside), the external microprocessor accesses the whole flash memory 205 (both the standard portion 205s and the local portion 205I) as usual.

[0027] Whenever the external microprocessor needs to perform generic data management operations involving the content of the local portion 205I of the flash memory, a Read/Write Buffer command is sent to the non-volatile memory device 120. In response thereto, the non-volatile memory device 120 is switched to the local mode of operation (making the local buffer 265 accessible from the outside).

[0028] One or more words are then written into the local buffer 265; each word is written into the location identified by the address ADR during a writing cycle, which is managed through corresponding control signals provided by the external microprocessor directly. In this way, the external microprocessor downloads information to the non-volatile memory device 120; the information consists of instructions (defining data management operations to be carried out by the embedded microprocessor 245) and possible corresponding data.

[0029] Once downloading of the information is completed, the external microprocessor sends a Start command to the non-volatile memory device 120. In response thereto, the embedded microprocessor 245 is started and the program code stored in the flash ROM 250 is run. The program code interprets the instructions loaded in the local buffer 265. These instructions will cause the embedded microprocessor 245 to carry out

corresponding operations on the data stored in the local buffer 265 and/or in the local portion 205I of the flash memory; for example, the data downloaded to the local buffer 265 is saved (either directly or after being processed by the embedded microprocessor 245) in the local portion 205I, or selected data is retrieved from the local portion 205I, processed by the embedded microprocessor 245 if necessary, and then stored into the local buffer 265.

[0030] As soon as processing is completed, the embedded microprocessor 245 updates the status register accordingly. The status register is periodically polled by the external microprocessor. When the external microprocessor detects completion of the processing, the external microprocessor reads the result of the processing stored in the local buffer 265 if necessary. Particularly, each word in the location of the local buffer 265 identified by the address ADR is read during a reading cycle, which is managed through corresponding control signals provided by the external microprocessor directly.

[0031] In the meanwhile, the external microprocessor is always allowed to access the standard portion 205s of the flash memory as usual. Moreover, if the external microprocessor needs to access the local portion 205I of the flash memory a Suspend command is sent to the non-volatile memory device 120. In response thereto, operation of the embedded microprocessor 245 is suspended (storing its current condition into the SRAM 255), and the status register is updated accordingly. The external microprocessor then sends a Word Read command to the non-volatile memory device 120, in order to read the words stored in the selected locations of the local portion 205I of the flash memory. Once reading of the local portion 205I of the flash memory has been completed, the external microprocessor sends a Resume command to the non-volatile memory device 120. In response thereto, operation of the embedded microprocessor 245 is resumed (after retrieving the stored condition from the SRAM 255), and the status register is restored accordingly.

[0032] In addition, the external microprocessor may force a hardware reset of the embedded microprocessor 245 by sending a Reset command to the non-volatile memory device 120. This command will cause operation of the embedded microprocessor to be aborted (with the status register that is updated accordingly).

[0033] The proposed architecture is used in different applications. For example, when the external microprocessor needs to update the content of the non-volatile device 120 the new data is downloaded to the local buffer 265; the local portion 205I of the flash memory is then updated accordingly under the control of the embedded microprocessor 245. Preferably, the embedded microprocessor manages a look-up table (stored in the SRAM 255), which defines a directory of the local portion 205I of the flash memory. In this way, the embedded microprocessor 245 emulates the behavior of a direct access bulk memory (such as a hard-disk).

[0034] Alternatively, the non-volatile memory device 120 is used to store compressed data. In this case, the (original) data is downloaded to the local buffer 265; the embedded microprocessor 245 directly controls compression of the data and its storing into the local portion 205I of the flash memory. Whenever the original data is required by the external microprocessor, the data is decompressed by the embedded microprocessor 245 and stored into the local buffer 265. The original data is then read from the local buffer 265 by the external microprocessor directly.

[0035] Similar considerations apply if the non-volatile memory device includes equivalent functional units, if the flash memory, the local buffer and/or the SRAM have a different capacity, if the embedded microprocessor is replaced with an equivalent programmable logic unit, if the status register and/or the configuration register are placed elsewhere, if the local buffer is implemented with equivalent means, if the standard portion and the local portion of the flash memory are configured in a different way, if a chip enable pin is used to actuates the local buffer (instead of the Read/Write Buffer command), if the status of the embedded microprocessor is monitored by hardware (for example, using a busy pin of the non-volatile memory device), if the non-volatile memory device supports other commands, if further functions are provided (for example, for preventing execution of conflicting operations on the flash memory). Alternatively, the local portion of the flash memory is used for different applications; for example, the embedded microprocessor controls encoding/decoding or ciphering/deciphering of data stored in the non-volatile memory device, the embedded microprocessor implements error correction procedures on the data stored in the flash memory, and the like.

[0036] More generally, the present invention proposes a non-volatile memory device including a flash memory. Means are provided for executing external commands; the external commands include a first subset of commands for accessing the flash memory directly. The memory device further includes a programmable logic unit and means for storing program code for the logic unit. The external commands includes a second subset of one or more commands for causing the logic unit to process information stored in at least one portion of the flash memory under the control of the program code.

[0037] The proposed architecture reduces the overload of any external processing unit employing the non-volatile memory device as a peripheral; in fact, most of the data management processing is carried out by the embedded microprocessor directly. In this way, the performance of a whole system (wherein the non-volatile memory device is included) is strongly increased.

[0038] The solution of the invention reduces the amount of data that must be transmitted between the non-volatile memory device and the external microprocessor. This non-volatile memory device is particularly advantageous in portable systems that are supplied by

a battery (such as mobile telephones), even if the use of the devised structure in different applications is not excluded.

[0039] In addition, any data management software already available for controlling and organizing the flash memory may be reused, and downloaded to the non-volatile memory device at production time.

[0040] The preferred embodiment of the invention described above offers further advantages.

[0041] Particularly, the non-volatile memory device includes a local buffer that is mapped on the local portion of the flash memory; the local buffer is directly accessible from the external microprocessor in the local mode of operation of the non-volatile memory device.

[0042] The proposed feature makes it possible to download information for the embedded microprocessor and to upload any result of the processing in a very simple manner.

[0043] Preferably, the flash memory is logically partitioned into a standard portion and a local portion (for the embedded microprocessor).

[0044] This feature ensures that the embedded microprocessor only accesses a dedicated portion of the flash memory.

[0045] Advantageously, the dimensions of the two portions of the flash memory are updated dynamically.

[0046] In this way, the flash memory may be configured for different conditions of use.

[0047] Alternatively, information is downloaded to and/or uploaded from the non-volatile memory device with another procedure (for example, only using the I/O buffer), the flash memory is partitioned in a different manner, the dimensions of the standard portion and of the local portion cannot be updated dynamically, the embedded microprocessor is allowed to access the whole flash memory, and the like.

[0048] As a further improvement, the program code controlling operation of the embedded microprocessor interprets instructions that have been loaded into the local buffer.

[0049] In this way, a number of operations may be requested to the embedded microprocessor employing a single command (for starting its operation).

[0050] Preferably, the non-volatile memory device supports two commands for suspending and for resuming, respectively, operation of the embedded microprocessor.

[0051] These commands allow the local portion of the flash memory to be accessed every moment.

[0052] Moreover, the non-volatile memory device includes means for providing an indication of the conclusion of the processing operations.

[0053] This feature makes it possible to synchronize operation of the embedded microprocessor and of the external microprocessor.

[0054] However, the solution of the present invention leads itself to be implemented controlling operation of the embedded microprocessor in a different way, with-

out any suspend and/or resume command, and even monitoring the processing operations in a different way.

[0055] In a preferred embodiment of the invention, the flash memory has a multiple banks architecture.

[0056] In this way, program code stored in the standard portion can be always read while updating data stored in the local portion.

[0057] Advantageously, the non-volatile memory device of the invention is integrated in a single chip of semiconductor material.

[0058] The proposed structure provides very high performance.

[0059] However, the non-volatile memory device of the invention is also suitable to be implemented with a different flash memory, or even distributing the components of the non-volatile memory device across two or more chips (but with the device that is always seen as a single peripheral from the outside).

[0060] Naturally, in order to satisfy local and specific requirements, a person skilled in the art may apply to the solution described above many modifications and alterations all of which, however, are included within the scope of protection of the invention as defined by the following claims.

Claims

1. A non-volatile memory device (120) including a flash memory (205) and means (225,227) for executing external commands, the external commands including a first subset of commands for accessing the flash memory directly,

characterized in that

the memory device further includes a programmable logic unit (245) and means (250) for storing program code for the logic unit, the external commands including a second subset of at least one command for causing the logic unit to process information stored in at least one portion (205l) of the flash memory under the control of the program code.

2. The memory device (120) according to claim 1, wherein the at least one portion (205l) of the flash memory is mapped on a corresponding address space, the memory device further including buffering means (265) mapped on said address space and means (220,225,245) for switching the memory device between a first operative condition and a second operative condition, the buffering means being non-accessible from the outside in the first operative condition and the buffering means being directly accessible from the outside instead of the at least one portion of the flash memory in the second operative condition.
3. The memory device (120) according to claim 1 or 2, further including means (248) for logically parti-

tioning the flash memory into a first (205s) and a second (205l) portion, the at least one portion of the flash memory consisting of the second portion.

4. The memory device (120) according to claim 3, further including means (220,225,245) for dynamically updating the dimensions of the first (205s) and the second (205l) portion of the flash memory.
5. The memory device (120) according to claim 3 or 4, wherein the information stored in the buffering means (265) includes instructions for causing the program code to execute corresponding processing operations on the information stored in the second portion (205l) of the flash memory, the second subset including a command for starting operation of the logic unit (245).
6. The memory device (120) according to claim 5, wherein the second subset further includes a command for suspending operation of the logic unit (245) and a command for resuming operation of the logic unit.
7. The memory device (120) according to any claim from 5 to 6, further including means (248) for providing an indication of the conclusion of the processing operations.
8. The memory device (120) according to any claim from 1 to 7, wherein the flash memory (205) has a multiple banks architecture.
9. The memory device (120) according to any claim from 1 to 8, wherein the memory device is integrated in a single chip of semiconductor material.
10. A data processing system (100) including the memory device (120) of any claim from 1 to 9.

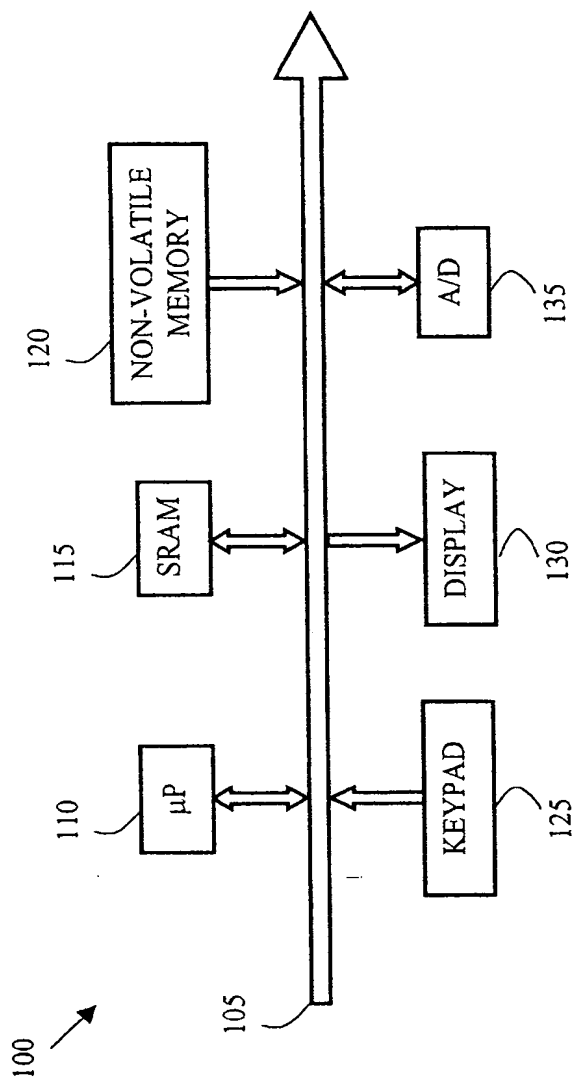


Fig.1

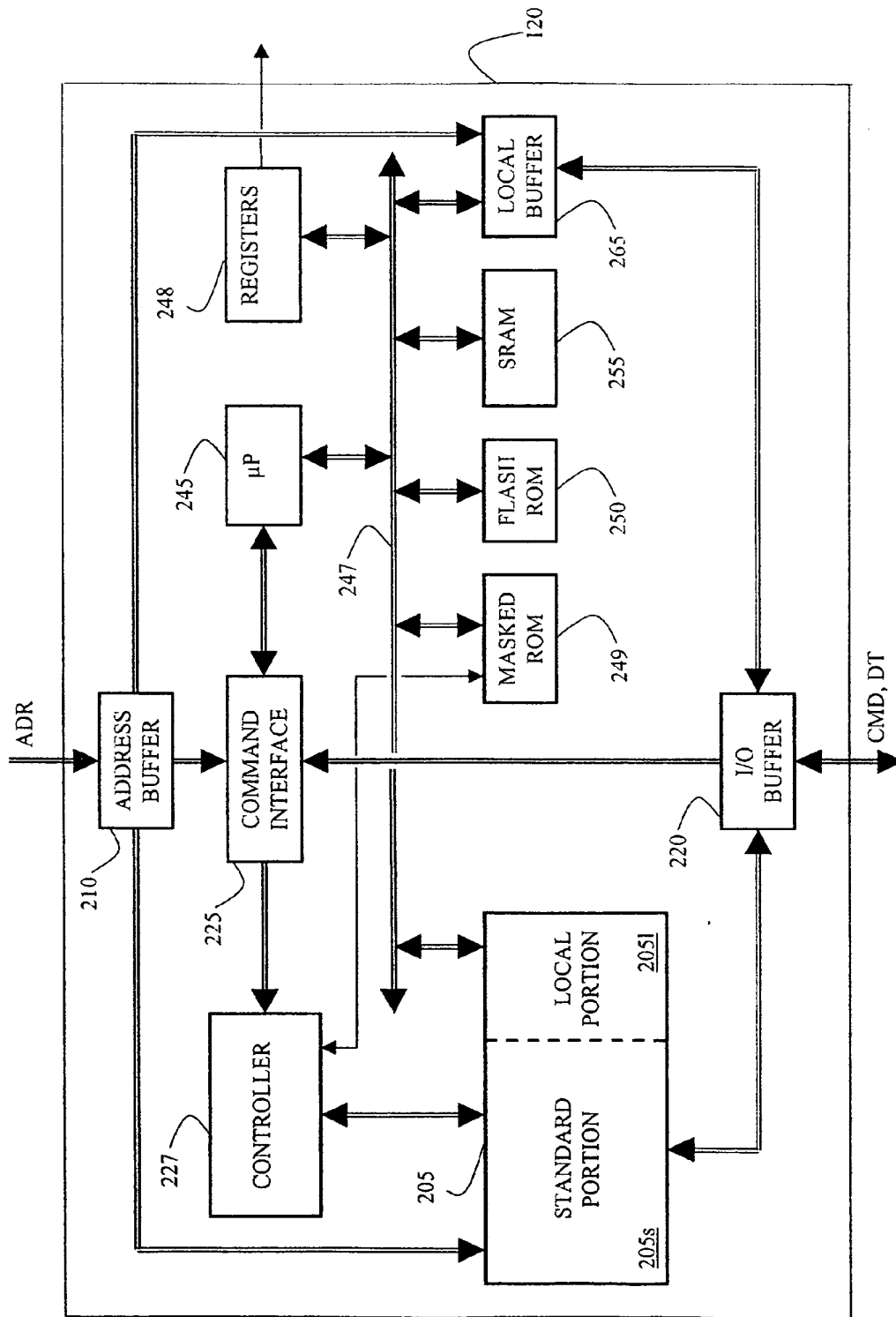


Fig.2



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 02 42 5152

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
Y	WO 98 29876 A (INTEL CORP) 9 July 1998 (1998-07-09) * page 12, line 3 - page 13, line 11; figure 2 * * page 14, line 6 - page 15, line 25; figure 3 * * page 19, line 29 - page 20, line 12 * * page 21, line 8 - page 22, line 6 *	1,8-10	611C16/10
Y	WO 98 28750 A (FUJITSU LTD ;ADVANCED MICRO DEVICES INC (US)) 2 July 1998 (1998-07-02) * page 4, line 21 - page 11, line 9; figures 1,2 *	1,8-10	
A	WO 01 45106 A (QUALCOMM INC) 21 June 2001 (2001-06-21) * abstract; figures 1,2 *	1,9,10	
A	FR 2 719 939 A (SGS THOMSON MICROELECTRONICS) 17 November 1995 (1995-11-17) * page 17, line 13 - page 20, line 29; figure 2 *	1,9,10	
A	US 5 463 757 A (FANDRICH MICKEY L ET AL) 31 October 1995 (1995-10-31) * abstract; figure 1 *	1	
A	US 6 240 019 B1 (SHIGA HITOSHI ET AL) 29 May 2001 (2001-05-29) * abstract; figure 5 *	1	
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 3 October 2002	Examiner Cummings, A
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document	

EPO FORM 1503 03.82 (p04c01)

**ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.**

EP 02 42 5152

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

03-10-2002

Patent document cited in search report		Publication date		Patent family member(s)	Publication date
WO 9829876	A	09-07-1998	US	5937423 A	10-08-1999
			AU	5146598 A	31-07-1998
			WO	9829876 A1	09-07-1998
WO 9828750	A	02-07-1998	US	5867430 A	02-02-1999
			DE	69707502 D1	22-11-2001
			DE	69707502 T2	13-06-2002
			EP	0944907 A1	29-09-1999
			JP	2000509871 T	02-08-2000
			TW	421798 B	11-02-2001
			WO	9828750 A1	02-07-1998
WO 0145106	A	21-06-2001	US	6407949 B1	18-06-2002
			AU	2271301 A	25-06-2001
			EP	1238393 A2	11-09-2002
			WO	0145106 A2	21-06-2001
			US	2001036109 A1	01-11-2001
FR 2719939	A	17-11-1995	FR	2719939 A1	17-11-1995
US 5463757	A	31-10-1995	FR	2672709 A1	14-08-1992
			JP	6028877 A	04-02-1994
US 6240019	B1	29-05-2001	JP	2000243095 A	08-09-2000
			US	2002080654 A1	27-06-2002
			US	6351417 B1	26-02-2002

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82